A Novel Approach for Mining and Fuzzy Simulation of Subnetworks From Large Biomolecular Networks

Xiaohua Hu, Bahrad Sokhansanj, Daniel Wu, and Yuchun Tang

***Abstract*—Understanding the biomolecular network implementing cellular function goes beyond the old dogma of "one gene: one function"; only through comprehensive system understanding can we predict the impact of genetic variation in the population, design effective disease therapeutics, and evaluate the potential side-effects of therapies. In this paper, we present a novel method to model the regulatory system that executes a cellular function, which can be represented as a biomolecular network. Our method consists of three steps. First, the biomolecular network is derived using data-mining approaches to extend the initial conceptual biomolecular network from the literature search, etc. Secondly, once the whole biomolecular network structure is complete, a novel scale-free network clustering approach is applied to obtain various subnetworks. Lastly, fuzzy rule based models are generated for the subnetworks and simulations are run to predict their behavior in the cellular context. The modeling results represent hypotheses that are tested against high-throughput data sets (microarrays and/or genetic screens) for both the natural system and perturbations. If computational results do not match experimental or previously published results, then new hypotheses are formed and they feed back into the data-mining and analyzing step to refine the biomolecuar network for the next iteration. This is repeated until a good match between modeling and data is obtained. Notably, the dynamic modeling component of this method depends on the automated network structure generation of the first component and the subnetwork clustering, which are both essential to make the solution tractable. Experimental results on human gene interaction networks and gene expression time series data for the human cell cycle indicate that our approach is promising for subnetwork mining and simulation from large biomolecular networks, as it produces a better convergence between continuous modeling and experiments.**

***Index Terms*—Biomedical literature mining, biomolecular network, fuzzy logic, information extraction, subnetwork.**

I. Introduction

WE are in the era of holistic biology. Massive amounts of biological data await interpretation. This calls for formal modeling and computational methods. In this paper, we present a method to model the regulatory system that executes a cellular function, which can be represented as a biomolecular network. Understanding the biomolecular network that implements cellular function goes beyond the old dogma of "one gene: one function"; only through comprehensive system understanding can we predict the impact of genetic variation in the population, design effective disease therapeutics, and evaluate the potential side-effects of therapies.

As biomolecular networks grow in size and complexity, biomolecular network models must become more rigorous to keep track of all the components and their interactions. This presents the need for computer simulation to manipulate and understand the biomolecular network model. However, a major challenge of modeling the dynamics of a biomolecular network is that conventional methods based on physical and chemical principles (such as systems of differential equations) require data that are difficult to accurately and consistently measure using either conventional or high-throughput technologies, which characteristically yield noisy, semiquantitative, and often relative data. For example, microarray gene expression ratios are ultimately obtained from pixel counts of relatively messy images [1]. Boolean networks (e.g., [2]) are computationally simple and do not depend on precise experimental data, and thus they are potentially suitable for handling both the complexity of biological networks and qualitative text-based data. However, Boolean models have been proven to lack the resolution needed to accurately model biomolecular interactions [3]. In contrast, various differential equation-based models (e.g., [4]) are computationally expensive and sensitive to imprecisely measured parameters (and virtually useless given purely qualitative data, i.e., from text-mining). Fuzzy logic [5] provides a mathematical framework that is compatible with poorly quantitative yet qualitatively significant data. Fuzzy logic is a natural language for linguistic modeling, thus it is consistent with the qualitative linguistic-graphical methods conventionally used to describe biological systems.

We present a hybrid approach that combines data mining and fuzzy modeling to build and analyze the biomolecular network of a cell process. It integrates the process of obtaining network structure directly with robust, fuzzy logic state-based dynamic simulation with qualitative (molecular biology) and noisy quantitative (biochemical) data to iteratively test and refine hypothetical biomolecular networks.

The dataflow of our method is illustrated in Fig. 1. The biomolecular network is derived by using data-mining approaches to extend the initial conceptual biomolecular network from the heterogeneous biodata sources, such as genome sequence homology, literature search, public microarray experiment databases, pathway databases, etc. Once the whole

Manuscript received March 22, 2006; revised October 2, 2006 and November 26, 2006. This work was supported in part by the National Science Foundation (NSF) under Career Grant NSF IIS 0448023 and NSF CCF 0514679 and in part by the Pennsylvania Department of Health under a Research Grant.

X. Hu and D. Wu are with the College of Information Science and Technology, Drexel University, Philadelphia, PA 19104 USA (e-mail: thu@cis.drexel.edu).

B. Sokhansanj is with the School of Biomedical Engineering, Science and Health Systems, Drexel University, Philadelphia, PA 19104 USA.

Y. Tang was with the Department of Computer Science, Georgia State University, Atlanta, GA 30302 USA. He is now with Secure Computing Corporation, Alpharetta, GA 30022 USA.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TFUZZ.2007.896248

biomolecular network structure is complete, a novel scale-free network clustering approach is applied to the biomolecular network to obtain various subnetworks. Then hypothetical fuzzy rule base models are generated for the subnetworks and simulate them to predict their dynamic biological behavior. The modeling results are verified against high-throughput data (microarrays and/or genetic screens) for both the natural system and perturbations. If computational results do not match experimental or previously published results, a new hypothesis is generated and fed back to the data-mining and analyzing step to refine the biomolecuar network for the next iteration, producing better convergence between continuous modeling and experiments. Notably, the dynamic modeling component of this method depends on the automated network structure generation of the first component and the subnetwork clustering, which are both essential to make the solution tractable.

The rest of this paper is organized as follows: In Section II, we review some of the related work in biomedical literature mining, community/subnetwork identification and fuzzy modeling for biomolecular networks. We present a novel algorithm, SNBuilder (Subnetwork Builder), in Section III for community structure analysis. The fuzzy modeling approach is discussed in Section IV along with experimental results. Section V concludes with our main findings and future research directions.

II. Related Work

In this section, we review some related work in biomedical literature mining, community structure analysis, and biomolecular networking modeling.

A. Biomedical Literature Mining

Biomedical literature mining, mainly from literature archived in the PubMed database, has attracted much attention recently from the information extraction, data mining, natural language understanding (NLP), and bioinformatics communities [6], [7]. Many methods have been proposed and various systems developed for extracting biological knowledge from biomedical literature, such as finding protein or gene names [8], [9], protein–protein interactions [10], protein–gene interactions [11], subcellular protein locations, functionality of genes, and protein synonyms [12]. For example, in its pioneering work in biomedical literature mining, [8] relies on special characteristics, such as the occurrence of uppercase letters, numerals, and special endings, to pinpoint protein names. Reference [9] extracts cooccurrences of gene names from Medline documents and uses them to predict their connections based on their joint and individual occurrence statistics. Reference [10] proposes an NLP–based approach to parse sentences in abstracts into grammatical units and then analyze sentences discussing interactions based on the frequency of individual words. Because of the complexity and variety of the English language, such an approach is inherently difficult. Reference [13] manually defines some regular expression patterns used to identify protein–protein interactions. The problem with that approach is that regular expression searches for abstracts containing relevant words, such as "interact," "bind," etc., poorly discriminate true hits from abstracts using the words in alternative senses and miss abstracts that use different language to describe the interactions. This approach may introduce many "false positives" or "false negatives," and it is unable to capture the new biological relationships not in those "manual" patterns.

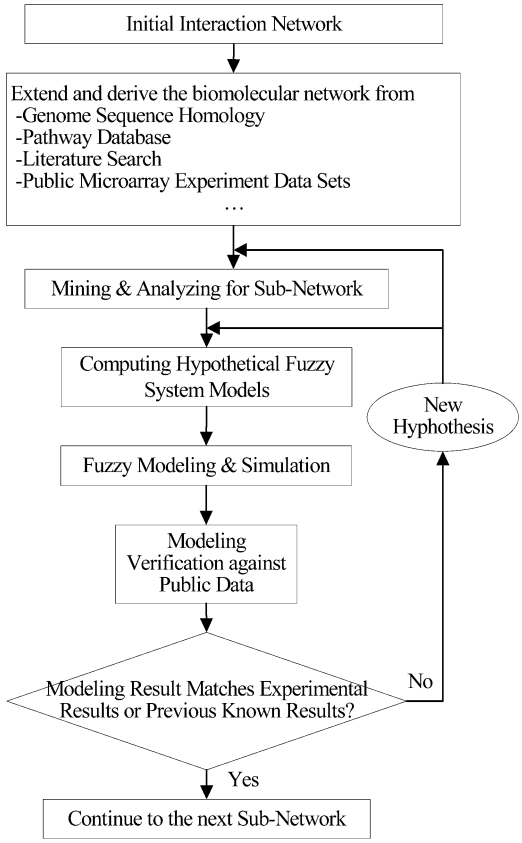

Fig. 1. Outline of mining and fuzzy modeling of biomolecular network.

Reference [14] proposes a Bayesian approach based on the frequencies of discriminating words found in the abstracts. Medline abstracts are scored for probability of discussing the topic of interest according to the frequencies of discriminating words found in the abstract. The highly likely abstracts are the sources for the curators for further examination for entry into the databases. Reference [15] develops MEDSYNDIKATE based on NLP techniques to extract knowledge from medical reports. Although the approaches differ, they can all be seen as examples of this process: first, what they will read is selected, important entities and relations between those entities are then identified, and finally this new information is combined with other documents and other knowledge. These systems, however, suffer from various weaknesses. First, the templates these systems are supplied with allow only factual information about particular entities chosen a priori (cell type, virus type, protein group, etc.), which are to be assembled from the analyzed documents. Also, these knowledge sources are considered to be entirely static. Accordingly, when the focus of interest of a user shifts to a topic not yet considered, new templates must be supplied or existing ones updated manually.

B. Community Structure Analysis

The study of community structure in a network is closely related to graph partitioning in graph theory and computer sci-

ence. It also has close ties with hierarchical clustering in sociology [16]. Recent years have witnessed intensive activity in this field, partly due to the dramatic increase in the scale of networks being studied. Because communities are believed to play a central role in the functional properties of complex networks [16], the ability to detect communities in networks could have practical applications. Studying the community structure of biological networks is of particular interest but is very challenging given the high data volume and the complex nature of interactions. In the context of biological networks, communities might represent structural or functional groupings. They can be synonymous with molecular modules, biochemical pathways, gene clusters, or protein complexes. Being able to identify the community structure in a biological network may help us to understand better the structure and dynamics of biological systems. Reference [17] develops an approach to growing genetic regulatory networks from seed genes. Their work is based on probabilistic Boolean networks, and subnetworks are constructed in the context of a directed graph using both the coefficient of determination and the Boolean function influence among genes. A similar approach is taken by [18] to find highly topically related communities in the Web based on the self-organization of the network structure and on a maximum flow method. Related works also include those that predict cocomplex proteins. Reference [19] uses a procedure integrating different data sources to predict the membership of protein complexes for individual genes based on two assumptions: that the function of any protein complex depends on the functions of its subunits and that all subunits of a protein complex share certain common properties. Reference [20] reports a molecular complex detection (MCODE) clustering algorithm that identifies molecular complexes in a large protein interaction network. MCODE is based on local network density—a modified measure of the clustering coefficient. Reference [21] uses a spectral analysis method to identify the topological structures, such as quasi-cliques and quasi-bipartites, in a protein–protein interaction (PPI) network. These topological structures are found to be biologically relevant functional groups. In our previous work, we developed a spectral-based clustering method using local density and vertex neighborhood to analyze the chromatin network [22], [23]. Two recent works along this line of research are based on the concept of network modularity introduced by [24]. The works in [25] and [26] both use computational analyses to cluster the yeast PPI network and discover that molecular modules are densely connected with each other but sparsely connected with the rest of the network.

C. Biomolecular Networking Modeling

A variety of approaches to state models have been implemented for gene and protein networks, including, among others, hidden Markov models [27], [28], Bayesian networks [29]–[31], linear neural networks [32], and finite state [33] and probabilistic Boolean networks [34], [35]. These and other methods are based on either treating biological variables at the crudest resolution (on or off in Boolean networks, with a few more levels possible for finite state models, but with rapidly growing complexity) or as absolute physical quantities. To integrate molecular biology data (generally linguistic and low-resolution), semiquantitative data (e.g., from microarrays), and quantitative data available for biomolecular networks, we propose to model the dynamics of the biomolecular network using fuzzy logic.

III. Analyzing the Biomolecular Network to Identify Subnetworks

In our previous work, we developed a novel scalable, portable, and robust system for extracting and mining data from biomedical literature: Bio-IEDM (biomedical information extraction and data mining) [36]. Bio-IEDM integrates information extraction and robust data mining to automatically extract and mine biological relationships from a huge collection of biomedical literature to help biologists in functional bioinformatics research.

Automated learning and consolidation tools such as Bio_IEDM serve several purposes.

1) They consolidate data about a single organism or a single class of entities (e.g., proteins, genes, etc.) in one place, allowing bioinformatics analysis on a global view of organisms at the molecular biology scale.
2) They make vast quantities of information searchable and manageable, as results are extracted in a structured format.
3) From this extracted knowledge they allow researchers to clarify biological relationships defined as "metadata" for hypothesis generation.

The procedure of extracting biological relationships from biomedical literature to construct a biomolecular network is discussed in our previous work [23], [36]; thus, in the rest of this section, we focus on mining the subnetworks of the biomolecular network derived from biomedical literature mining.

The interpretation of large-scale protein network data depends on our ability to identify significant substructures (communities) in the data, a computationally intensive task. Many algorithms for detecting community structure in networks have been proposed. They can be roughly classified into two categories: divisive and agglomerative. The divisive approach recursively removes vertices (or edges) until the network is separated into its components or communities, whereas the agglomerative approach starts with isolated individual vertices and joins together small communities. One important algorithm is proposed by Girvan and Newman (the GN algorithm) [37]. The GN algorithm is based on the concept of "betweenness," a quantitative measure of the number of shortest paths passing through a given vertex (or edge). The GN algorithm detects communities in a network by recursively removing these high-betweenness vertices (or edges). It has produced good results and is well adopted by different authors in studying various networks [16]. However, it has a major disadvantage, which is its computational cost. For sparse networks with n vertices, the GN algorithm is of $\mathrm{O}(n^3)$ time. Various alternative algorithms have been proposed [38]–[41] that attempt to improve either the quality of the community structure or the computational efficiency. As discussed in [42], edge-betweenness uses properties calculated from the whole graph, allowing information from nonlocal features to be used in the clustering. The edge-betweenness algorithm does not scale well to larger

graphs, currently making this method most appropriate for studies focused on specific areas of the proteome.

The goal of this paper is to address a slightly different question about the community structure in a PPI network, i.e., what is the community to which a given protein (or proteins) belongs? We are motivated by two main factors. First, due to the complexity and modularity of biological networks, it is more feasible computationally to study a community containing a small number of proteins of interest. Secondly, sometimes the whole community structure of the network may not be our primary concern. Rather, we may be more interested in finding the community that contains a protein (or proteins) of interest. Our aim is to discover relatively small subnetworks such that proteins inside the subnetwork interact significantly and, meanwhile, are not strongly influenced by proteins outside the subnetwork. Subnetworks are constructed starting with a seed consisting of one or more proteins believed to participate in a viable subnetwork. Functionalities and regulatory relationships among seed proteins may be partially known or may simply be of interest. Given the seed, we iteratively adjoin new proteins following an adapted definition of a community in a network. The subnetworks built from our models may provide valuable theoretical guidance for experiment.

A. The Algorithm SNBuilder (Subnetwork Builder)

We intuitively model the protein–protein interaction network as an undirected graph, where vertices represent proteins and edges represent interactions between pairs of proteins. An undirected graph $G = (V, E)$ is composed of two sets: vertices V and edges E. An edge e is defined as a pair of vertices (u, v) denoting the direct connection between vertices u and v. The graphs we use in this paper are undirected, unweighted, and simple, meaning there are no self-loops or parallel edges.

For a subgraph $G' \subset G$ and a vertex i belonging to G', we define the in-community degree for vertex $i, k_i^{\text{in}}(G')$, to be the number of edges connecting vertex i to other vertices belonging to G' and the out-community degree $k_i^{\text{out}}(G')$ to be the number of edges connecting vertex i to other vertices that are in G but do not belong to G'.

In our algorithm, we adopt the quantitative definitions of community defined in [43], i.e., the subgraph G' is a community in a strong sense if $k_i^{\text{in}}(G') > k_i^{\text{out}}(G')$ for each vertex i in G' and in a weak sense if the sum of all degrees within G' is greater than the sum of all degrees from G' to the rest of the graph.

The algorithm, called *SNBuilder,* accepts the seed protein s, gets the neighbors of s, finds the core of the community to build, and expands the core to find the eventual community.

The two major components of *SNBuilder* are *FindCore* and *ExpandCore*. In fact, *FindCore* (lines 8–14) performs a naïve search for maximum clique in the neighborhood of the seed protein by recursively removing vertices with the lowest in-community degree until either 1) all vertices in the core set have the same in-community degree ($K_{\min} = K_{\max}$, i.e., the resulting subgraph is a clique), or 2) all vertices except the seed have the same in-community degree (a star-like structure).

The algorithm performs a breadth first expansion in the core expanding step. It first builds a candidate set containing the core and all vertices adjacent to each vertex in the core (line 16). A candidate vertex will then be added to the core if it meets one of the following conditions (line 21): 1) its in-community degree is greater than its out-community degree, i.e., the quantitative definition of community in a strong sense ($k_t^{\text{in}} > k_t^{\text{out}}$), or 2) its affinity coefficient is greater than or equals the affinity threshold f.

We define the affinity coefficient of a vertex to a network as the fraction of its in-community degree over the size of the network excluding the vertex itself ($k_t^{\text{in}}(D)/(|D| - 1)$). We introduce the affinity coefficient and the affinity threshold f to provide a degree of relaxation when expanding the core because it is too strict to require every expanding vertex to be a strong sense community member. Even though a candidate vertex may not have an in-community degree larger than out-community degree, it may connect to all (or even most) other members of the network, indicating a strong tie between the candidate vertex and the network. We use an affinity threshold f of one in our implementation, meaning that in order to be eligible to add to the core set, the candidate vertex has to connect to all other vertices in the core set. However, f may be relaxed to be less than one if necessary or so desired.

In addition, a distance parameter (d) is provided to restrict how far away a candidate vertex to the seed can be considered eligible for expansion. Quite often, a given seed may not always situate in the center of the resulting subnetwork. The distance parameter serves as the shortest path threshold to ensure that all members of the obtained subnetwork will be within specified proximity to the seed. A large enough value of d, such as one that is larger than the longest path from the seed to all other vertices in the network, will virtually lift this distance restriction.

FindCore is a heuristic search for a maximum complete subgraph in the neighborhood N of seed s. Let K be the size of N; then the worst case running time of *FindCore* is $O(K^2)$. The *ExpandCore* part costs, in the worst case, approximately $|V| + |E|+$ overhead. $|V|$ accounts for the expanding of the core; at most, all vertices in V, minus what are already in the core, would be included. $|E|$ accounts for calculating the in- and out-degrees for the candidate vertices that are not in the core but are in the neighborhood of the core. The overhead is caused by recalculating the in- and out-degrees of neighboring vertices every time the *FindCore* is recursively called. The number of these vertices is dependent on the size of the community we are building and the connectivity of the community to the rest of the network but not the overall size of the network. For biological networks, the graphs we deal with are mostly sparse and small world; therefore, the running time of our algorithm is close to linear.

Algorithm 1 SNBuilder(G, s, f, d)

1: $G(V, E)$ is the input graph with vertex set V and edge set E.

2: s is the seed vertex; f is the affinity threshold; d is the distance threshold.

3: $N \leftarrow \{\text{Adjacency list of } s\} \cup \{s\}$

4: $C \leftarrow \text{FindCore}(N)$

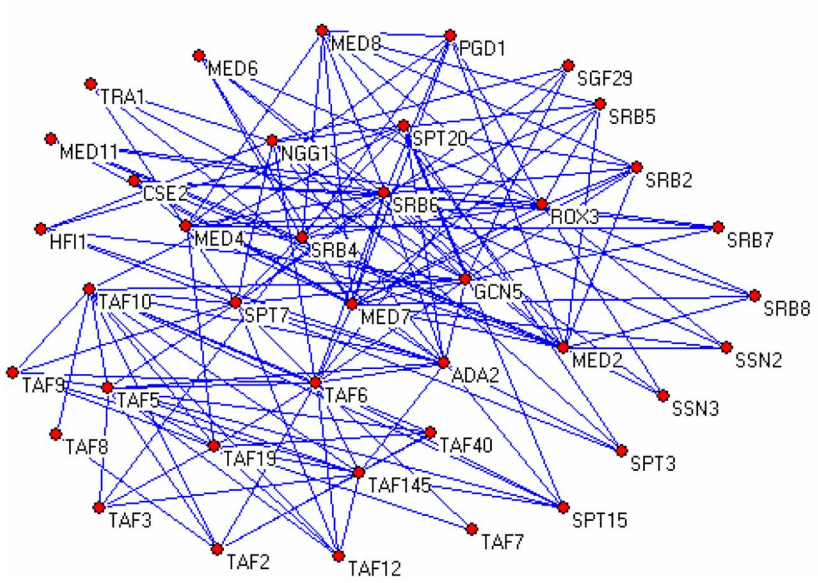

Fig. 2. SAGA/SRB community.

5: $C' \leftarrow \text{ExpandCore}(C, f, d)$

6: **return** C'

7: FindCore(N)

8: **for each** $v \in N$

9: calculate $k_v^{\text{in}}(N)$

10: **end for**

11: $K_{\min} \leftarrow \min\{k_v^{\text{in}}(N), v \in N\}$

12: $K_{\max} \leftarrow \max\{k_v^{\text{in}}(N), v \in N\}$

13: **if** $K_{\min} = K_{\max}$ or $(k_i^{\text{in}}(N) = k_j^{\text{in}}(N), \forall i, j \in N, i, j \neq s, i \neq j)$ **then return** N

14: **else return** FindCore($N - \{v\}, k_v^{\text{in}}(N) = K_{\min}$)

15: ExpandCore(C, f, d)

16: $D \leftarrow \cup_{(v,w)\in E, v\in C, w\notin C}\{v, w\}$

17: $C' \leftarrow C$

18: **for each** $t \in D, t \notin C$, and distance$(t, s) <= d$

19: calculate $k_t^{\text{in}}(D)$

20: calculate $k_t^{\text{out}}(D)$

21: **if** $k_t^{\text{in}}(D) > k_t^{\text{out}}(D)$ **or** $k_t^{\text{in}}(D)/|D|f$ **then** $C' \leftarrow C' \cup \{t\}$

22: **end for**

23: **if** $C' = C$ **then return** C

24: **else return** ExpandCore(C', f, d)

B. Evaluation of SNBuilder

Because there is no alternative approach to our method, we decide to compare the performance of our algorithm to the work on predicting protein complex membership by [44], which reported results of queries with four complexes using probabilistic network reliability (we will refer to their work as the PNR method in the following discussion). Four communities are identified by SNBuilder using one protein as seed from each of the query complexes used by the PNR method. The seed protein is selected randomly from the "core" protein set. The figures for visualizing the identified communities are created using Pajek [45]. The community figures are extracted from the network we build using the above-mentioned data set with out-of-community connections omitted. The proteins in each community are annotated with a brief description obtained from the MIPS complex catalogue database. As a comparison, we use Complexpander, an implementation of the PNR method [44],[1] to predict cocomplex using the core protein set that contains the same seed protein used by SNBuilder. For all our queries when using Complexpander, we select the option to use the MIPS complex catalogue database. We record the ranking of the members in our identified communities that also appear in the cocomplex candidate list predicted by Comlexpander.

An example of such a community identified by SNBuilder, using TAF6 as seed, is shown in Fig. 2. TAF6 is a component of the SAGA complex which is a multifunctional coactivator that regulates transcription by RNA polymerase II [46]. The SAGA complex is listed in the MIPS complex catalogue as a known cellular complex consisting of 16 proteins. As shown in Table I, the community identified by our algorithm contains 39 members, including 14 of the 16 SAGA complex proteins listed in MIPS (indicated by an asterisk in the *Alias* column). The community also contains 14 of 21 proteins listed in MIPS as Kornberg's mediator (SRB) complex. The rest of the proteins in

[1]Available at http://llama.med.harvard.edu/Software.html.

TABLE I
SAGA/SRB COMMUNITY

Protein	Alias	Description	Rank
YDR448w	ADA2*	general transcriptional adaptor or co-activator	1
YNR010w	CSE2†	subunit of RNA polymerase II mediator complex	
YGR252w	GCN5*	histone acetyltransferase	2
YPL254w	HFI1*	transcriptional co-activator	3
YMR112c	MED11†	mediator complex subunit	
YDL005c	MED2†	transcriptional regulation mediator	20
YOR174w	MED4†	transcription regulation mediator	23
YHR058c	MED6†	RNA polymerase II transcriptional regulation mediator	
YOL135c	MED7†	member of RNA Polymerase II transcriptional regulation mediator complex	21
YBR193c	MED8†	transcriptional regulation mediator	24
YDR176w	NGG1*	general transcriptional adaptor or co-activator	10
YGL025c	PGD1†	mediator complex subunit	37
YBL093c	ROX3†	transcription factor	
YCL010c	SGF29*	SAGA associated factor	43
YER148w	SPT15	the TATA-binding protein TBP	15
YOL148c	SPT20*	member of the TBP class of SPT proteins that alter transcription site selection	4
YDR392w	SPT3*	general transcriptional adaptor or co-activator	13
YBR081c	SPT7*	involved in alteration of transcription start site selection	5
YHR041c	SRB2†	DNA-directed RNA polymerase II holoenzyme and Kornberg^s mediator (SRB) subcomplex subunit	
YER022w	SRB4†	DNA-directed RNA polymerase II holoenzyme and Kornberg^s mediator (SRB) subcomplex subunit	27
YGR104c	SRB5†	DNA-directed RNA polymerase II holoenzyme and Kornberg^s mediator (SRB) subcomplex subunit	
YBR253w	SRB6†	DNA-directed RNA polymerase II suppressor protein	19
YDR308c	SRB7†	DNA-directed RNA polymerase II holoenzyme and kornberg^s mediator (SRB) subcomplex subunit	46
YCR081w	SRB8	DNA-directed RNA polymerase II holoenzyme and Srb10 CDK subcomplex subunit	
YDR443c	SSN2	DNA-directed RNA polymerase II holoenzyme and Srb10 CDK subcomplex subunit	
YPL042c	SSN3	cyclin-dependent CTD kinase	
YGR274c	TAF1	TFIID subunit (TBP-associated factor), 145 kD	14
YDR167w	TAF10*	TFIID and SAGA subunit	7
YML015c	TAF11	TFIID subunit (TBP-associated factor), 40KD	18
YDR145w	TAF12*	TFIID and SAGA subunit	8
YML098w	TAF13	TFIID subunit (TBP-associated factor), 19 kD	17
YCR042c	TAF2	component of TFIID complex	22
YPL011c	TAF3	component of the TBP-associated protein complex	50
YBR198c	TAF5*	TFIID and SAGA subunit	9
YGL112c	TAF6*	TFIID and SAGA subunit	
YMR227c	TAF7	TFIID subunit (TBP-associated factor), 67 kD	
YML114c	TAF8	TBP Associated Factor 65 KDa	
YMR236w	TAF9*	TFIID and SAGA subunit	11
YHR099w	TRA1*	component of the Ada-Spt transcriptional regulatory complex	12

Proteins that belong to SAGA complex listed in MIPS complex catalogue database are indicated by (*) and those belonging to SRB complex are indicated by (†). Ranking is done by running Complexpander using the seed (taf6) as the core protein set.

the community are either TATA-binding proteins, transcription factor IID (TFIID) subunits, or SRB related. TFIID is a complex involved in initiation of RNA polymerase II transcription. SAGA and TFIID are structurally and functionally correlated, make overlapping contributions to the expression of RNA polymerase II transcribed genes. SRB complex is a mediator that conveys regulatory signals from DNA-binding transcription factors to RNA polymerase II [47]. In addition, 27 of the top 50 potential cocomplex proteins (nine of the top ten) predicted by Complexpander, not including the seed proteins, are in the identified community.

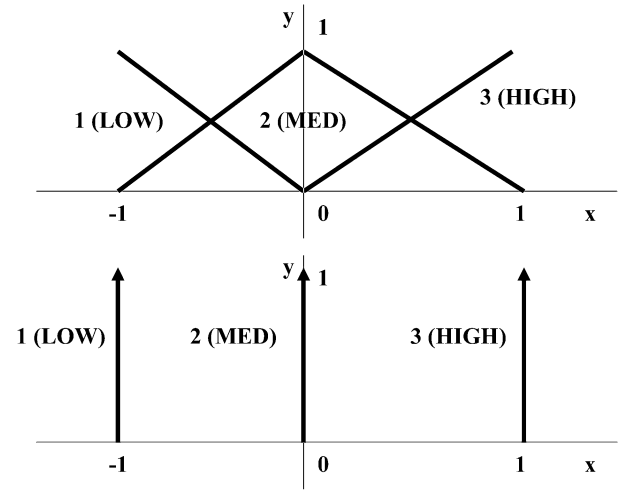

Fig. 3. Fuzzification (conversion from quantity to fuzzy set; top) and defuzzification (conversion back from fuzzy set to quantity; bottom) schemes, Defuzzification (centroid method, point set definitions [48]) is equivalent to dividing the difference between memberships in HIGH and LOW by the sum of memberships in all sets.

IV. FUZZY LOGIC STATE-BASED SIMULATION OF THE BIOMOLECULAR NETWORK

It is difficult, if not impossible, to perform experiments that can obtain continuous, dynamic information required to develop a complete physical spatiotemporal model for the dynamics of the biomolecular network. The greatest barrier is due to the cost (time, technology, risk of experimental failure) of doing multiple experiments; thus, in most cases, time series data are undersampled. Often, only one time point is taken during a transition in system state—for example, at a set time following a perturbation when it is thought (based on other information or cruder experiments) that system response is at its peak or has reached a quasi-steady state. If complete information for system behavior is unavailable, a state model can be developed and applied to make predictions about how the system will respond to perturbations, as well to identify and evaluate hypotheses for state transition mechanisms.

Fuzzy logic is a generalization of Boolean logic that allows for a continuum of set membership between 0 (absolutely "false") and 1.0 (absolutely "true") [5]. Fuzzy logic allows for "linguistic" rule-based modeling, an extension of Boolean if/then rules using English words to represent fuzzy states of variables (described in detail for engineering applications in [48] and citations within). Through defined "fuzzification" and "defuzzification" functions, the quantity of a variable can be translated into a fuzzy membership in multiple sets that are typically given names like "LOW," "MEDIUM," "HIGH," etc. Thus, a set of rules can be written as, for example, "if input is LOW then output is MEDIUM, if input is MEDIUM then output is LOW, if input is HIGH then output is HIGH." This allows for the construction of rule-based models similar to qualitative rules found in biological literature, i.e., "if repressor gene A is expressed at a LOW level then the expression of its target gene is HIGH," and variations thereof.

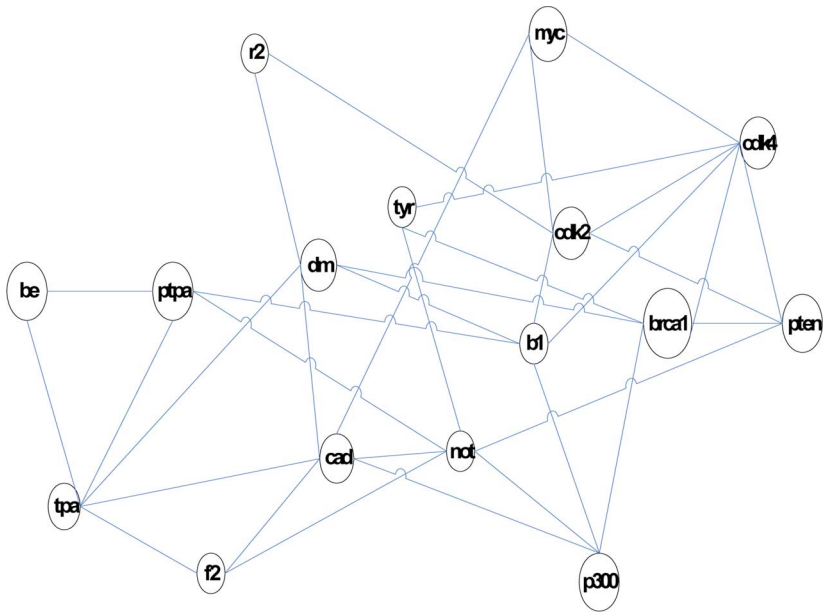

Fig. 4. A sample of subnetwork.

Concretely, Fig. 3 describes (top) the mapping of quantities on an interval $[-1, 1]$ to continuous [0,1] membership in fuzzy sets used for translating semiquantitative data to fuzzy set language and (bottom) the mapping of membership in fuzzy sets back to a "crisp" numerical value. These definitions were chosen primarily to prevent error arising from fuzzification and defuzzification for monotonic linear relationships. To apply this fuzzy set definition scheme requires determining a suitable normalization of semiquantitative values to the interval $[-1, 1]$. Previous fuzzification schemes for gene expression involved normalizing by the maximum expression level [49] and designing different schemes for each variable [50]. Our current approach, outlined in [51], is to take the normalized arctangent of the base 2 logarithm of the expression ratios (base 2 logarithms are a common way of symmetrizing ratiometric data for gene expression found in comparative microarray experiments). This general normalization scheme can be flexibly applied to any ratiometric gene expression data set, as the arctangent function is defined over an infinite domain and reflects the general tendency of microarray experiments to saturate at high and low expression levels. Using arctangent normalization and the fuzzification of Fig. 3, we previously generated semiquantitative and qualitatively accurate fuzzy models for the yeast cell cycle gene network [51].

A. *Fuzzy Model Identification*

The fuzzy biomolecular network model is a set of rule sets for each node (in this case gene) in the network, which govern the response to each fuzzy state of the input genes to that node (the output gene). Examples of such rule sets are given below. The result for each input gene is summed to give the overall behavior of the output gene, resulting in a linear fuzzy model that has been called the "union rule configuration" (URC) [52]. URC fuzzy logic has been applied previously to modeling biological systems, such as the *lac* operon of E. coli [50]. While simply summing rules in the URC does not allow for nonlinear interactions between inputs, these may be represented either through the introduction of artificial "hidden layers" analogous to those in neural network models [53] or by including more details to the network, using additional information to add the necessary layers for a relevant linear network model.

To evaluate the accuracy and feasibility of fuzzy biomolecular network modeling, we considered the gene network that corresponded to a subnetwork found using the automated data mining and subsequent clustering methods. It involves human genes related to p53, apoptosis, DNA damage response, and cell cycle. The automated method replaces manual curating, which is time-consuming, inefficient, and potentially susceptible to the bias of the modelers and biologists in identifying putative connections between genes. The subnetwork clustering method is essential to reduce the thousands of genes implicated in these processes to an exemplar, representative set, allowing the modeling and analysis of simulation results to be tractable problems.

Fuzzy rule sets are generated for genes in the network. Edges in Fig. 4 are taken to represent potential connections between genes, defining the set of possible inputs to each gene. Following previously described and tested combinatorial URC fuzzy rule search procedure [51], we exhaustively generate all possible rule combinations for the inputs on each gene (including the "null" rule, equivalent to excluding a potential rule). The procedure of [51] is modified to allow only those input genes that are identified by mining literature and clustering the resulting biomolecular network map (i.e., as shown, for example, in Fig. 4). An additional assumption we make is that if an expression level of an input gene is MEDIUM it must result in an output expression level of MEDIUM also. An example of a possible rule is "If Input is LOW then Output is LOW, if Input is MED then Output is MED, if Input is HIGH then Output is MED"). Three fuzzy sets are used to retain tractability of the rule search method, which examines

TABLE II
BEST FITTING RULES FOR FIVE INPUT GENES

	Protein Name	CDK2	MYC	CDK4	EP300	BRCA 1
DMTF	dm	---	---	---	---	LMH
BRCA1	brca1	---	---	0	MML	---
H1FX	h1	0	---	LMH	MMH	---
HE	he	---	---	---	---	---
PPP2R4	ptpa	LMH	---	---	---	---
MYC	myc	MMH	---	LMH	---	---
NR4A2	not	---	---	---	MMH	---
F2	f2	---	---	---	---	---
PTEN	pt	MML	---	0	---	LMH
RRM2	r2	LMH	---	---	---	---
PLAT	tpa	---	---	---	---	---
TYR	tyr	---	---	HML	---	LMM
CAD	cad	---	LMH	---	LMM	---
CDK2	cdk2	---	MMH	LMH	---	---
CDK4	cdk4	LMM	LMM	---	---	0
EP300	p300	---	---	---	---	HMM
Train E		0.5257	0.8454	0.7318	0.8794	0.6267
Test E		0.6387	0.6825	0.5294	1.2104	1.9369

all potential hypotheses consistent with the data. However, this still represents a significant advance in resolution over Boolean logic because of the nonbinary membership in LOW and HIGH fuzzy sets. As shown in our previous work [51], this method has the potential to generate predicted gene expression time series that are quantitatively consistent with experimental data.

We apply the resulting fuzzy rules to microarray gene expression time series data for the human cell cycle, shown in [54]. Such cell cycle microarray data have been criticized for being very noisy, and previous methods for identifying cell cycle–regulated genes may be artifacts (i.e., see [55], which focuses on a similar data set). However, these problems are endemic to genomic and proteomic measurement methods. Thus, the human cell cycle data set is a reasonable test for the practical application of fuzzy logic modeling on data for which conventional methods can fail. We evaluated each fuzzy rule set from the exhaustive search at each time point in the data set and compared the prediction and experimental data for each gene using an error metric (E) that emphasizes the correlation in qualitative expression changes between predicted and experimental data

$$E = \frac{\sum_{j=1}^{M} (x_j - \tilde{x}_j)^2}{\sum_{j=1}^{M} (x_j - \bar{x})^2} \tag{1}$$

where the M experimental data $\{x_j\}$ (with mean $\bar{x}$) and defuzzified predicted numerical values $\{\tilde{x}_j\}$ for the output gene.

Table II shows the best fitting rules for five genes. Each entry in the table is the rule for the input gene (rows) acting on the output gene (columns), where "HML" denotes the rule: "If Input is LOW, then Output is HIGH; if Input is MED, then Output is MED; if Input is HIGH then Output is LOW." A dashed line in Table II means that the gene is not an input in the biomolecular network of Fig. 4, and a "0" means that the best fit rule excludes one of the potential inputs in the network. Notably, Table II shows the genes that encode the proteins in Fig. 4. This results in some differences in terminology; for example, EP300 encodes the protein p300. Also, where aliases for gene names exist, the more common usage is given in Table II.

There are independent data [54] for five methods of cell cycle synchronization, two of which are complete for the genes in the subnetwork we studied. One data set ("Thy-Thy 3") was used as the "training set," and for each gene exhaustively generated rule sets were ranked based on the error (E) of that rule set on the data (the error of the best fit rules is given in the row of Table II labeled "Train E"). The rules were then simulated at each time point in the "test set," a different kind of cell cycle synchronization ("Thy-Noc"); these are the errors in the last row of Table II. As shown in Fig. 5, agreement between the predictions of the best fit rule on the training set and the data in the test set is excellent in some cases, in particular CDK2 and CDK4, which are known to be cell cycle–regulated genes (and thus are expected to have a regular pattern of behavior in this data set). In others (e.g., BRCA1), there is little agreement. These patterns are reflected when looking at overall results for the exhaustive rule search.

Fig. 6 plots the error on the test set against the error on the training set for every possible rule for two genes. A linear trend indicates that rules that have a low fit error in the training set tend also to have a relatively low fit error on the test set, and vice versa for poorly fitting data. Fig. 6 shows two outcomes: CDK4, which shows excellent agreement between fuzzy rules generated on the training set and predictions for the test set $(\mathrm{R}^2 = 0.74)$ and the relatively noisy results for BRCA1 (no apparent trend, though some particularly poorly fitting rules on the training set also fit the test data poorly).

V. CONCLUSION AND DISCUSSION

In this paper, we present a new method for adaptive modeling of biomolecular networks. The biomolecular network model of cell function comprises gene and protein expression, interaction, and regulation. The method iteratively mines and organizes quantitative and qualitative data to generate scalable hypothetical biomolecular network structures. The dynamics of these computational hypotheses are tested and refined through cycles of fuzzy logic based simulation and laboratory experiments. Our method addresses three major challenges of modeling cell function and regulation—heterogeneous information sources, hypothetical and contradictory biological information, and scalability. Biomolecular data is derived from many sources, including noisy and qualitative experiments, sequences, structures, microarrays, mass spectra, and narrative text. We use fuzzy logic methods, previously developed for gene networks, as a robust and general representation for heterogeneous quantitative, qualitative, and linguistic biomolecular data. While only microarray data are presented as an example in this paper, the fuzzy logic framework of

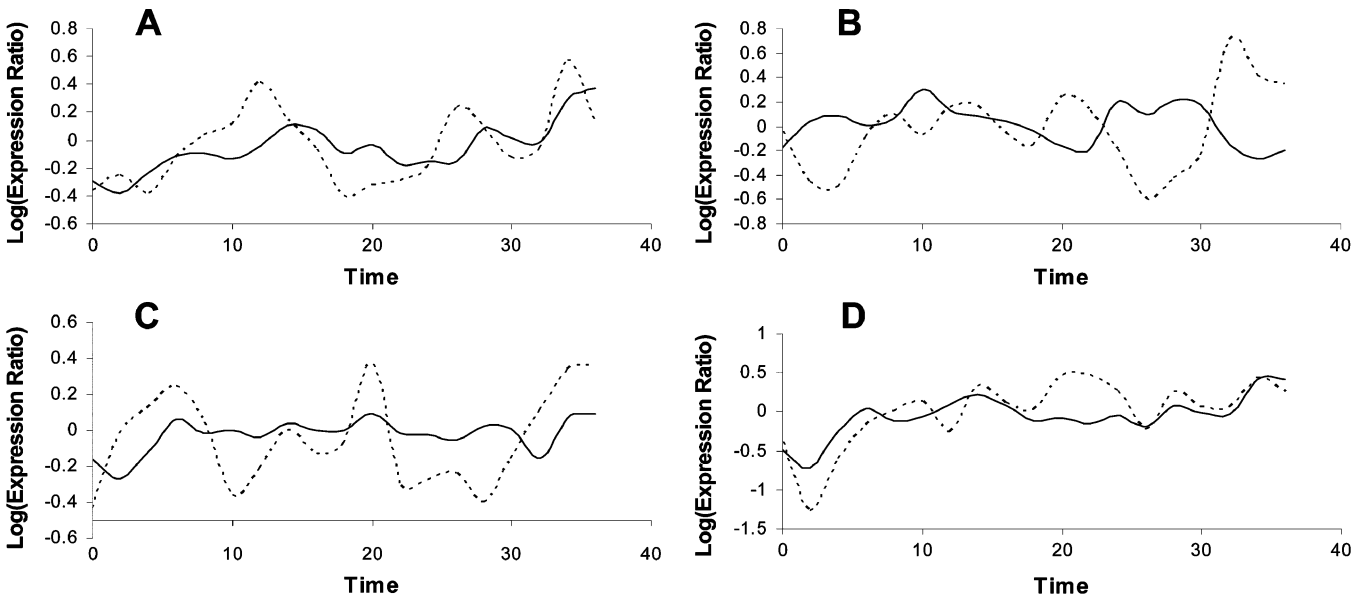

Fig. 5. Best fit rule on training set "Thy-Thy 3" (as shown in Table II) predicting gene expression on the test data set (solid line) compared to actual data from the test set "Thy-Noc" (dashed line) for (A) CDK2, (B) BRCA1, (C) EP300, and (D) CDK4.

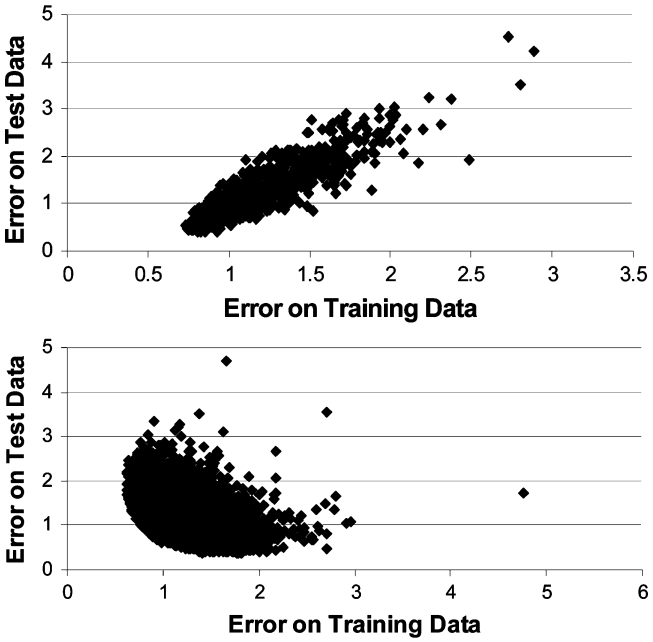

Fig. 6. Error on test data against error on training data for each rule combination exhaustively generated for (top) CDK4 and (bottom) BRCA1.

representing biomolecular expression states can be easily extended to protein and metabolite levels. This is a key point, as gene networks are an abstraction representing only one aspect of biomolecular networks and they must be integrated with protein–protein interaction networks and metabolite profiling to develop a comprehensive portrait of cellular function.

In general, the fuzzy logic method allows for inconsistencies and potentially noisy data to be identified and used to generate alternative computational hypotheses for biomolecular networks. The method is tractable and scalable because novel clustering methods are applied to adaptively extract biologically significant subnetworks for simulation and hypothesis testing. Fuzzy simulation of hypothetical biomolecular network models is compared with experimental data to select and refine plausible hypotheses. We combine the simulation result with the computationally derived metamodel to identify key genes whose perturbation would generate the data set that could most optimally differentiate between the alternative biomolecular network hypotheses. Consequently, by uniting the system identification and simulation components of the modeling procedure into an integrated method, we can develop a cyclical flow of information that moves from modeling, through experiments, through updates to be included in the global biological knowledge base, which then feeds back into modeling. Such a flow is designed specifically to respond to the challenges of designing and interpreting high-throughput experiments, which can, in the future, evolve in concert with modeling and information management.

REFERENCES

[1] J. P. Fitch and B. Sokhansanj, "Genomic engineering: Moving beyond DNA sequence to function," *Proc. IEEE*, vol. 88, no. 12, pp. 1949–1971, Dec. 2000.

[2] P. D'haeseleer, S. Liang, and R. Somogyi, "Genetic network inference: From co-expression clustering to reverse engineering," *Bioinformatics*, vol. 16, no. 8, pp. 707–726, Apr. 2000.

[3] L. Glass and S. A. Kauffman, "The logical analysis of continuous nonlinear biochemical control: Networks," *J. Theor. Biol.*, vol. 39, no. 1, pp. 103–129, Apr. 1973.

[4] J. Tegnér, M. K. S. Yeung, J. Hasty, and J. J. Collins, "Reverse engineering gene networks: Integrating genetic perturbations with dynamical modeling," *Proc. Nat. Acad. Sci. USA*, vol. 100, no. 10, pp. 5944–5949, May 2003.

[5] L. A. Zadeh, "Fuzzy sets," *Inf. Contr.*, vol. 8, no. 3, pp. 338–353, Jun. 1965.

[6] L. Hirschman, J. C. Park, J. Tsujil, L. Wong, and C. H. Wu, "Accomplishments and challenges in literature data mining for biology," *Bioinformatics*, vol. 18, no. 12, pp. 1553–1561, Dec. 2002.

[7] T. Hasegawa, S. Sekine, and R. Grishman, "Discovering relations among named entities from large corpora," in *Proc. Annu. Meeting Assoc. Comput. Linguist.*, Barcelona, Spain, Jul. 21–26, 2004, pp. 415–422.

[8] K. Fukuda, T. Tsunoda, A. Tamura, and T. Takagi, "Toward information extraction: Identifying protein names from biological papers," in *Proc. Pac. Symp. Biocomput.*, Jan. 1998, pp. 707–718.

[9] B. J. Stapley and G. Benoit, "Biobibliometrics: Information retrieval and visualization from co-occurrences of gene names in medline abstracts," in *Proc. Pac. Symp. Biocomput.*, Jan. 2000, pp. 529–540.

[10] C. Blaschke, M. A. Andrade, C. Ouzounis, and A. Valencia, "Automatic extraction of biological information from scientific text: Protein-protein interactions," in *Proc. Int. Conf. Intell. Syst. Mol. Biol.*, Heidelberg, Germany, Aug. 6–10, 1999, pp. 60–67.

[11] J. H. Chiang and H. H. Yu, "MeKE: Discovering the functions of gene products from biomedical literature via sentence alignment," *Bioinformatics*, vol. 19, no. 11, pp. 1417–1422, Jul. 2003.

[12] B. de Brujin and J. Martin, "Literature mining in molecular biology," in *Proc. .EFMI Workshop Natural Lang.*, Nicosia, Cyprus, Mar. 8–9, 2002, pp. 1–5.

[13] T. Ono, H. Hishigaki, A. Tanigami, and T. Takagi, "Automated extraction of information on protein-protein-interactions from the biological literature," *Bioinformatics*, vol. 17, no. 2, pp. 155–161, Feb. 2001.

[14] E. Marcotte, I. Xenarios, and D. Eisenberg, "Mining literature for protein interactions," *Bioinformatics*, vol. 17, no. 4, pp. 359–363, Apr. 2001.

[15] U. Hahn, M. Romacker, and S. Schulz, "Creating knowledge repositories from biomedical reports: The MEDSYNDIKATE text mining system," in *Proc. Pac. Symp. Biocomput.*, Jan. 2002, pp. 338–349.

[16] M. E. J. Newman, "The structure and function of complex networks," *SIAM Rev.*, vol. 45, no. 2, pp. 167–256, 2003.

[17] R. F. Hashimoto, S. Kim, I. Shmulevich, W. Zhang, M. L. Bittner, and E. R. Dougherty, "Growing genetic regulatory networks from seed genes," *Bioinformatics*, vol. 20, no. 8, pp. 1241–247, May 2004.

[18] G. W. Flake, S. R. Lawrence, C. L. Giles, and F. M. Coetzee, "Self-organization and identification of web communities," *IEEE Computer*, vol. 35, pp. 66–71, Mar. 2002.

[19] R. Jansen, N. Lan, J. Qian, and M. Gerstein, "Integration of genomic datasets to predict protein complexes in yeast," *J. Struct. Funct. Genomics*, vol. 2, no. 2, pp. 71–81, 2002.

[20] G. D. Bader and C. W. V. Hogue, "An automated method for finding molecular complexes in large protein interaction networks," *BMC Bioinformatics*, vol. 4, no. 2, Jan. 2003.

[21] D. Bu, Y. Zhao, L. Cai, H. Xue, X. Zhu, H. Lu, J. Zhang, S. Sun, L. Ling, N. Zhang, G. Li, and R. Chen, "Topological structure analysis of the protein-protein interaction network in budding yeast," *Nucl. Acids Res.*, vol. 31, no. 9, pp. 2443–2450, May 2003.

[22] X. Hu, "Mining and analysing scale-free protein-protein interaction network," *Int. J. Bioinformatics Res. Applicat.*, vol. 1, no. 1, pp. 81–101, 2005.

[23] X. Hu, I. Yoo, I.-Y. Song, M. Song, J. Han, and M. Lechner, "Extracting and mining protein-protein interaction network from biomedical literature," in *Proc. 2004 IEEE Symp. Comput. Intell. Bioinformatics Comput. Biol.*, San Diego, CA, Oct. 7–8, 2004, pp. 244–251.

[24] L. H. Hartwell, J. J. Hopfield, S. Leibler, and A. W. Murray, "From molecular to modular cell biology," *Nature*, vol. 402, no. 6761 Suppl., pp. C47–C52, Dec. 1999.

[25] V. Spirin and L. A. Mirny, "Protein complexes and functional modules in molecular networks," *Proc. Nat. Acad. Sci. USA*, vol. 100, no. 21, pp. 12123–12128, Oct. 2003.

[26] A. W. Rives and T. Galitski, "Modular organization of cellular networks," *Proc. Nat. Acad. Sci. USA*, vol. 100, no. 3, pp. 1128–1133, Feb. 2003.

[27] R. A. Rosales, M. Fill, and A. L. Escobar, "Calcium regulation of single ryanodine receptor channel gating analyzed using HMM/MCMC statistical methods," *J. Gen. Physiol.*, vol. 121, pp. 533–553, 2004.

[28] A. Schliep, A. Schonhuth, and C. Steinhoff, "Using hidden Markov models to analyze gene expression time course data," *Bioinformatics*, vol. 19, no. Suppl. 1, pp. i255–i263, Jul. 2003.

[29] C. Rangel, J. Angus, Z. Ghahramani, M. Lioumi, E. Sotheran, A. Gaiba, D. L. Wild, and F. Falciani, "Modeling T-cell activation using gene expression profiling and state-space models," *Bioinformatics*, vol. 20, no. 9, pp. 1361–1372, Jun. 2004.

[30] M. J. Beal, F. Falciani, Z. Ghahramani, C. Rangel, and D. L. Wild, "A Bayesian approach to reconstructing genetic regulatory networks with hidden factors," *Bioinformatics*, vol. 21, no. 3, pp. 349–356, Feb. 2004.

[31] I. Nachman, A. Regev, and N. Friedman, "Inferring quantitative models of regulatory networks from expression data," *Bioinformatics*, vol. 20, no. Suppl 1, pp. i248–i256, Aug. 2004.

[32] P. D'Haeseleer, X. Wen, S. Fuhrman, and R. Somogyi, "Linear modeling of mRNA expresion levels during CNS development and injury," in *Proc. Pac. Symp. Biocomput.*, Jan. 1999, pp. 41–52.

[33] R. Laubenbacher and B. Stigler, "A computational algebra approach to the reverse-engineering of gene regulatory networks," *J. Theor. Biol.*, vol. 229, no. 4, pp. 523–537, Aug. 2004.

[34] I. Shmulevich, E. R. Dougherty, S. Kim, and W. Zhang, "Probabilistic boolean networks: A rule-based uncertainty model for gene regulatory networks," *Bioinformatics*, vol. 18, no. 2, pp. 261–274, Feb. 2002.

[35] T. J. Perkins, M. Hallett, and L. Glass, "Inferring models of gene expression dynamics," *J. Theor. Biol.*, vol. 230, no. 3, pp. 289–299, Oct. 2004.

[36] X. Hu and D. Wu, "Data mining and predictive modeling of biomolecular network from biomedical literature databases," *IEEE/ACM Trans. Comp. Biol. Bioinf.*, to be published.

[37] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proc. Nat. Acad. Sci. USA*, vol. 99, no. 12, pp. 7821–7826, June 2002.

[38] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Phys. Rev. E*, vol. 69, no. 2, 026113, Feb. 2004.

[39] M. E. J. Newman, "Fast algorithm for detecting community structure in networks," *Phys. Rev. E*, vol. 69, no. 6, 066133, Jun. 2004.

[40] L. Donetti and M. A. Munoz, "Detecting network communities: A new systematic and efficient algorithm," *J. Stat. Mech.*, no. 10, p. 10012, Oct. 2004.

[41] S. White and P. Smyth, "A spectral clustering approach to finding communities in graph," in *Proc. SIAM Int. Conf. Data Mining*, Newport Beach, CA, Apr. 2005.

[42] P. Holme, M. Huss, and H. Jeong, "Subnetwork hierarchies of biochemical pathways," *Bioinformatics*, vol. 19, no. 4, pp. 532–538, Mar. 2003.

[43] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisi, "Defining and identifying communities in networks," *Proc. Nat. Acad. Sci. USA*, vol. 101, no. 9, pp. 2658–2663, Mar. 2004.

[44] S. Asthana, O. D. King, F. D. Gibbons, and F. P. Roth, "Predicting protein complex membership using probabilistic network reliability," *Genome Res.*, vol. 14, no. 6, pp. 1170–1175, Jun. 2004.

[45] V. Batagelj and A. Mrvar, "Pajek: Program for large network analysis," *Connections*, vol. 21, pp. 47–57, 1998.

[46] P. Y. Wu, C. Ruhlmann, F. Winston, and P. Schultz P, "Molecular architecture of the S. cerevisiae SAGA complex," *Mol. Cell*, vol. 15, no. 2, pp. 199–208, Jul. 2004.

[47] B. Guglielmi, N. L. van Berkum, B. Klapholz, T. Bijma, M. Boube, C. Boschiero, H. M. Bourbon, F. C. P. Holstege, and M. Werner, "A high resolution protein interaction map of the yeast mediator complex," *Nucl. Acids Res.*, vol. 32, no. 18, pp. 5379–5391, Oct. 2004.

[48] J. Mendel, "Fuzzy logic systems for engineering: A tutorial," *Proc. IEEE*, vol. 83, no. 3, pp. 345–377, Mar. 1995.

[49] B. A. Sokhansanj, J. B. Garnham, and J. P. Fitch, "Interpreting data from microarray experiments to build models of microbial genetic regulation networks," *Proc. SPIE*, vol. 4623, pp. 27–37, Jan. 2002.

[50] B. A. Sokhansanj, G. R. Rodrigue, and J. P. Fitch JP, "Building and testing scalable fuzzy models of bacterial regulation," in *Proc. 2002 Int. Conf. Comput. Nanosci. Nanotechnol.*, San Juan, Puerto Rico, Apr. 22–25, 2002.

[51] B. A. Sokhansanj and D. M. Wilson 3rd, "Oxidative DNA damage background estimated by a system model of base excision repair," *Free Rad. Biol. Med.*, vol. 37, no. 3, pp. 422–427, Aug. 2004.

[52] W. E. Combs and J. E. Andrews, "Combinatorial rule explosion eliminated by a fuzzy rule configuration," *IEEE Trans. Fuzzy Syst.*, vol. 6, pp. 1–11, Feb. 1998.

[53] J. J. Weinschenk, R. J. Marks II, and W. E. Combs, "Layered URC fuzzy systems: A novel link between fuzzy systems and neural networks," in *Proc. 2003 Int. Joint Conf. Neural Netw.*, Portland, OR, Jul. 20–24, 2003.

[54] M. L. Whitfield, G. Sherlock, A. J. Saldanha, J. I. Murray, C. A. Ball, K. E. Alexander, J. C. Matese, C. M. Perou, M. M. Hurt, P. O. Brown, and D. Botstein, "Identification of genes periodically expressed in the human cell cycle and their expression in tumors," *Mol. Biol. Cell.*, vol. 13, pp. 1977–2000, 2002.

[55] K. Shedden and S. Cooper, "Analysis of cell-cycle-specific gene expression in human cells as determined by microarrays and double-thymidine block synchronization," *Proc. Nat. Acad. Sci. USA*, vol. 99, no. 7, pp. 4379–4384, Apr. 2002.

Xiaohua Hu received the B.Sc. degree in software from Wuhan University, China, in 1985, the M.Eng. degree in computer engineering from the Institute of Computing Technology, Chinese Academy of Science, China, in 1988, the M.Sc. degree in computer science from Simon Fraser University, Burnaby, BC, Canada, in 1992, and the Ph.D. degree in computer science from the University of Regina, Regina, SK, Canada 1995.

He is an Assistant Professor and Founding Director of the Data Mining and Bioinformatics Lab, College of Information Science and Technology, Drexel University, Philadelphia, PA. From 1994 to 1998, he was a Research Scientist in data mining with Nortel Network Research Center, GTE Labs (now Verizon Labs). He has worked on many projects related to data mining for real-time telephone switch system diagnosis, data management, and wireless churn prediction. Among them, the Churn Analysis, Modeling and Prediction project (CHAMP) was nominated for GTE's highest technical achievement award in 1997. From 1998–2002, he designed and developed data-mining commercial software for various startup companies (KSP, Blue Martini Software), founded the company DMW software, and successfully deployed a few data-mining products/systems to Fortune 100 companies such as Chase, Citibank, Sprint for credit fraud detection, e-personalization, and customer management systems. His current research interests are in biomedical literature data mining, bioinformatics, text mining, semantic Web mining and reasoning, rough set theory and application, information extraction, and information retrieval. He has published more than 130 peer-reviewed research papers in various journals, conferences, and books. He has coedited eight books/proceedings. He has been a Program Cochair/Conference Cochair of nine international conferences/workshops and a Program Committee Member in more than 40 international conferences in the above areas. He is Founding Editor-in-Chief of the *International Journal of Data Mining and Bioinformatics* and an Associate Editor/Editorial Board Member of four international journals. His research projects are supported by the National Science Foundation, U.S. Department of Education, and Pennsylvania Department of Health.

Prof. Hu received the 2005 National Science Foundation Career award, the Best Paper Award at the 2004 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology, the 2006 IEEE Granular Computing Outstanding Service Award, and the 2001 IEEE Data Mining Outstanding Service Award. He is the Founding Advisory Board Member and Secretary of the IEEE Granular Computing Task Force.

Bahrad Sokhansanj received the B.S. degree in engineering physics from the University of Saskatchewan, Saskatoon, SK, Canada, and the M.S. and Ph.D. degrees in applied science from the University of California, Davis-Livermore.

He pursued postdoctoral work at Lawrence Livermore National Laboratory. He is an Assistant Professor in the School of Biomedical Engineering, Science, and Health Systems, Drexel University, Philadelphia, PA. At Drexel, he leads the Molecular Health Engineering Laboratory, which develops quantitative experimental biology methods in conjuction with novel analysis and modeling methods. The focus of the laboratory is on a systems approach to a critical problem in chronic disease, including cancer: the impact of chronic inflammation on cell repair, apoptosis, and death in acute infections.

Daniel Wu received the B.S. degree in biochemistry from Xiamen University, China, and the M.S. degree in physiology and in computer science from Pennsylvania State University, University Park, in 1996 and 2001, respectively. He is currently pursuing the Ph.D. degree in the College of Information Science and Technology, Drexel University, Philadelphia, PA.

His research interests are in data mining, bioinformatics, and biomolecular network analysis.

Yuchun Tang received the B.S. degree from the Civil Aviation University of China, Tianjin, in 1996, the M.S. degree from Beijing Institute of Technology, Beijing, China, in 1999, and the Ph.D. degree in computer science from Georgia State University, Atlanta, in 2006.

From 1999 to 2001, he was a Research Scientist with the Institute of Computer Science and Technology, Beijing University. During summer 2003, he was a Research Visitor with the Berkeley Initiative in Soft Computing program, University of California at Berkeley. From 2004 to 2005, he was a Research Fellow with the Molecular Basis for Disease (MBD) program, Georgia State University. He is currently a Research Scientist with Secure Computing Corporation (formerly CipherTrust Inc.), Alpharetta, GA. His research interests include knowledge discovery and data mining, machine learning, statistical learning, computational intelligence, soft computing, granular computing, text mining, artificial intelligence, intelligent data analysis, and decision support systems.